

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Arrouye, et al.

U.S. Serial No.: Unassigned

Filing Date: Herewith

Title: Method And Apparatus For
"Just-In-Time" Dynamic Loading
And Unloading of Computer
Software Libraries

Examiner: Unassigned


Group Art Unit No.: Unassigned

CERTIFICATE OF MAILING (37 C.F.R. § 1.8)

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Label No. EL 144 928 145 US in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date: August 30, 2001

Typed or printed name Jennifer Pask

Signature



Date

August 30, 2001

PRELIMINARY AMENDMENTAssistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

Please enter the following amendments and remarks prior to consideration of the present application.

In the Specification:

Please amend the Specification as follows:

On page 1, lines 8 - 9, please delete "No. _____
(corresponding to attorney docket number P2206)" and insert therefor
--No. 09/161,758, issued as U.S. Patent No. 6,230,310 --.

On page 1, lines 11 - 12, please delete "No. _____
(corresponding to attorney docket number P2211)" and insert therefor
-- No. 09/162,125--.

On page 1, lines 15 - 16, please delete "No. _____
(corresponding to attorney docket number P2210)" and insert therefor
-- No. 09/162,126--.

On page 11, line 15, delete "is equal to" and insert therefor -- is less
than--.

On page 11, line 18, delete "greater than" and insert therefor --greater
than or equal to--.

In the Claims:

Please cancel Claims 1 - 23, 38 - 51 without prejudice and without
disclaimer.

Please amend Claims 24 - 37 as follows:

24. (amended) A computer system for dynamically and automatically
loading and unloading a software library [in]to and from memory
in a computer, said software library including one or more
library routines and capable of being [used]referenced by an
application or other software module, said computer system
comprising:

one or more library structures, each library structure
corresponding to a software library, each library structure
including a library implementation module containing code for
implementing the corresponding software library and a library
loader containing entry points corresponding to entry points of
the corresponding software library and code for loading and
unloading the corresponding library implementation module; and

[an operating system or other]software for performing the
following steps:

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library[containing a library routine], said loading occurring [just] prior to when [said]a library routine of the software library is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

automatically unloading from memory the library implementation module after the execution of the library routine is completed, wherein the software library is unloaded automatically without an explicit unload request from the application or other software module.

26. (amended) The computer system as defined in Claim 24 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module.
27. (amended) A computer system for dynamically and automatically loading and unloading a software library [in]to and from memory in a computer, said software library including one or more library routines and capable of being [used]referenced by an

application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

[an operating system or other]software for performing the following steps:

loading into memory a library implementation module for a software library[containing a library routine], said loading occurring [just] prior to when [said]a library routine of the software library is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory and scheduled to be unloaded from memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

automatically scheduling the unloading of the library implementation module; and

unless the scheduled unload has been cancelled, automatically unloading the library implementation module as scheduled, wherein the software library is scheduled for unload and unloaded automatically without an explicit unload request from the application or other software module.

29. (amended) The computer system as defined in Claim 27 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module.
30. (amended) The computer system as defined in Claim 27 wherein said step of scheduling the unloading of the library implementation module includes delaying the unloading of the library for a specified delay and unloading the library implementation module upon completion of the delay[setting an associated criteria and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use].
31. (amended) The computer system as defined in Claim 30 wherein said [associated criteria]specified delay is a time delay.
34. (amended) The computer system as defined in Claim 30 wherein said [associated criteria]specified delay is based on system resources.
36. (amended) The computer system for dynamically and automatically loading and unloading a software library [in]to and from memory in a computer, said software library capable of being [used]referenced by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

[an operating system or other]software for performing the following steps:

_____ loading a software library into memory[just prior to or contemporaneous with execution of a library routine in said software library]; and

_____ automatically unloading said software library from memory after completion of execution of said library routine, wherein the software library is unloaded automatically without an explicit unload request from the application or other software module.

37. (amended) The computer system of Claim 36 wherein said unloading of said software library comprises the steps of:

scheduling the unloading of the library implementation module;

setting an associated criteria;

checking whether said set associated criteria has been met;
and

unloading the library implementation module if after said associated criteria has been met, said library implementation module is not [in use]being referenced by an application or other software module.

Please add new Claims 52 – 75 as follows:

52. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said step of scheduling the unloading of the library implementation module includes setting a time delay and unloading the library implementation module if after said time delay has been met, said library implementation module is not in use.

53. (new) The computer system as defined in Claim 52 wherein said time delay is fixed.

54. (new) The computer system as defined in Claim 52 wherein said time delay is variable.

55. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is

to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria based on system resources and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.

56. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria, unsetting an associated criteria if said scheduled unloading is cancelled, and resetting the associated criteria if the library implementation module is again scheduled for unloading.

57. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure

including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria, unsetting an associated criteria if said scheduled unloading is cancelled and resetting the associated criteria if the library implementation module is again scheduled for unloading.

58. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading a software library into memory just prior to or contemporaneous with execution of a library routine in said software library; and

unloading said software library from memory after completion of execution of said library routine, wherein said unloading of said software library includes scheduling the unloading of the library implementation module, setting an associated criteria, checking whether said set associated criteria has been met, and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.

59. (new) A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:

determining one or more software libraries which are potentially needed during execution of an application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.

60. (new) A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:

determining one or more software libraries which are potentially needed during execution of an application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading including determining whether the library implementation module is in memory, if the library implementation module is not in memory, then loading the library implementation module, and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said step of scheduling the unloading of the library implementation module includes setting a time delay and unloading the library implementation module if after said time delay has been met, said library implementation module is not in use.

61. (new) The computer-readable medium as defined in Claim 60 wherein said time delay is fixed.
62. (new) The computer-readable medium as defined in Claim 60 wherein said time delay is variable.
63. (new) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an application or other software module, said computer system comprising:

one or more software libraries, each software library containing one or more library routines, each software library having a library loader and a library implementation module, the library loader specifying entry points corresponding to entry points in the software library and the library implementation module containing computer code to implement the software library, the library loader handling the step of unloading the software library;

means for loading into memory a library loader for each potentially needed library, a library loader for a library being loaded by the time the application or other software module executes a routine in that library;

means for loading into memory a library implementation module for a software library, said loading occurring prior to when a library routine of the software library is executed by the application or other software module, said loading handled by the library loader for the software library;

means for determining whether any application or other software module is referencing the software library; and

means for automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module, wherein the software library is unloaded automatically without an explicit unload request from the application or other software module.

64. (new) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an application or other software module, said computer system comprising:

one or more software libraries, each software library containing one or more library routines

means for determining whether any application or other software module is referencing the software library; and

means for automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module,

wherein the software library is unloaded automatically without an explicit unload request from the application or other software module, wherein said means for automatically unloading the software library comprises means for delaying the unloading of the software library for a specified delay and means for unloading the software library upon completion of the delay.

65. (new) The computer system as defined in claim 64, wherein said means for automatically unloading the software library further comprises:

means for determining at completion of the delay whether the software library is being referenced by an application or other software module; and

means for unloading the software module only if the software library is not being referenced by an application or other software module;

66. (new) The computer system as defined in claim 64 further comprising:

when an application or other software module references or otherwise invokes a routine in a software library, means for determining whether the software library is subject to a delayed unloading; and

if the software library is subject to a delayed unloading, means for canceling the delayed unloading.

67. (new) The computer system as defined in Claim 64 wherein said specified delay is a time-based delay.

68. (new) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an

application or other software module, said computer system comprising:

a processor;

a memory;

a disk;

one or more software libraries stored on said disk, each software library containing one or more library routines;

software operated on by said processor to perform the following steps:

determining whether any application or other software module is referencing the software library; and

automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module, wherein the software library is unloaded automatically without an explicit unload request from the application or other software module.

69. (new) The computer system as defined in claim 68 wherein said software library is comprised of a library loader and a library implementation module, the library loader specifying entry points corresponding to entry points in the software library and the library implementation module containing computer code to implement the software library, the library loader handling the step of unloading the software library, said software performing the further steps of:

loading into memory a library loader for each potentially needed library, a library loader for a library

being loaded by the time the application or other software module executes a routine in that library; and

loading into memory a library implementation module for a software library, said loading occurring just prior to when a library routine of the software library is executed by the application or other software module, said loading handled by the library loader for the software library.

70. (new) The computer system as defined in claim 68 wherein said step of automatically unloading the software library comprises the steps of delaying the unloading of the software library for a specified delay and unloading the software library upon completion of the delay.

71. (new) The computer system as defined in claim 68, said step of automatically unloading the software library further comprises the steps of:

determining at completion of the delay whether the software library is being referenced by an application or other software module; and

unloading the software module only if the software library is not being referenced by an application or other software module;

72. (new) The computer system as defined in claim 70 further comprising the steps of:

when an application or other software module references or otherwise invokes a routine in a software library, determining whether the software library is subject to a delayed unloading; and

if the software library is subject to a delayed unloading, cancelling the delayed unloading.

73. (new) The computer system as defined in Claim 70 wherein said specified delay is a time-based delay.

74. (new) A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:

loading a software library into memory; and
automatically unloading said software library from memory after completion of execution of said library routine by delaying the unloading for a delay period and canceling a delayed unloading of a software library if a library routine of the software library is referenced by an application or other software module during the delay period.

75. (new) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.

REMARKS

Claims 1 - 51 are currently pending. Applicants herein cancel Claims 1 - 23 and 38 - 51 without prejudice and without disclaimer and present new Claims 52 - 75.

Applicants amend the Specification herein to provide the serial numbers and patent numbers, if applicable, of corresponding referenced applications. Also, Applicants amend the Specification herein to correct minor errors.

Among other things, the present invention as claimed herein provides for "just-in-time" automatic dynamic loading and unloading of software libraries. (Specification, page 3, lines 8 - 10 and page 10, lines 4 - 36). A service requester or other client software does not need to issue an explicit unload or load request of a particular library. (page 10, lines 4 - 36) Rather, the loading and unloading of a library is handled automatically by the library loader for that library. (page 10, lines 4 - 36).

Moreover, according to an aspect of the invention, the unloading of a software library is delayed to avoid the unloading of a library which will soon be needed again. Among other things, this is particularly useful, for example, when executing a series of calls to the same library. (Specification, page 12, line 4 - page 16, line 2). The delay imposed between scheduling a library for unloading and actually unloading the library may be based on criteria, including, but not limited to, time duration and/or system resources, e.g. process scheduling or deferred execution. (Specification, page 13, lines 7 - 34). In an aspect of the invention, the unloading is delayed for a fixed pre-determined time period, thereby allowing subsequent calls to that library to occur within that time period without unloading and reloading of the library.

Applicants believe that amended Claims 24 – 37, as well as new Claims 52 – 75 are in condition for allowance and respectfully request that the Examiner enter the amendments and new Claims herein and grant allowance of the present application.

Respectfully submitted,

APPLE COMPUTER, INC.

Date: August 30, 2001



Helene Plotka Workman
Attorney for Applicant
Reg. No. 35,981
1 Infinite Loop, M/S: 3-PAT
Cupertino, CA 95014
(408) 974-5081 – direct dial
(408) 974-5436 – fax